

# Spyral Notebooks as a Supplement to Voyant Tools<sup>1</sup>

Paper by Kaylin Land<sup>2</sup>, Geoffrey Rockwell<sup>3</sup>, and Andrew MacDonald<sup>1</sup>

## Abstract

This paper introduces Spyral Notebooks, a notebook environment that extends Voyant Tools and offers a space for both notating and presenting analysis and for developing JavaScript code that extends Voyant. In the paper we present the design justifications for Spyral, showcase some of the analytical possibilities of Spyral, and address recent criticism of notebook environments. We suggest Spyral Notebooks help teach students to think-through digital text analysis and argue that the design of Spyral makes it rich in potential for the humanities researcher. We also highlight the collaborative possibilities of Spyral and show ways in which Spyral Notebooks have been used for collaborative digital humanities projects. Finally, we discuss future plans for Spyral.

Dans cet article nous présentons le Spyral Notebooks, un notebook ou calepin électronique qui étend le Voyant Tools et offre un cadre pour le commentaire et la présentation de l'analyse et pour le développement de code JavaScript. Nous justifions la conception pour Spyral, soulignons des possibilités analytiques de Spyral et discutons des critiques de calepins électroniques. Nous suggérons que le Spyral Notebooks enseigne aux étudiants entendre l'analyse de texte numérique et nous disputons que la conception de Spyral rend prometteur pour les chercheurs humanistes. Nous soulignons aussi les possibilités de collaboration avec Spyral Notebooks. Finalement, nous discutons les directions futures pour Spyral.

---

<sup>1</sup> Spyral was initially developed by Stéfan Sinclair with support from the Text Mining the Novel SSHRC Partnership led by Andrew Piper. It is now supported by the LINC project funded by CFI and led by Susan Brown.

<sup>2</sup> McGill University

<sup>3</sup> University of Alberta

## Introduction

Voyant Tools is widely recognized for being user-friendly, intuitive and useful for teaching.<sup>4</sup> The online digital text analysis tool site offers researchers, students, and amateur users alike the ability to analyze textual materials using a suite of different analytical tools. However, Voyant offers limited options for sharing, saving, and building upon analysis done in the browser application. This paper introduces Spyral Notebooks, a notebook environment that extends Voyant Tools and offers a space for both notating and presenting analysis and for developing JavaScript code that extends Voyant. In the paper we present the design justifications for Spyral, showcase some of the analytical possibilities of Spyral, and address recent criticism of notebook environments. We suggest Spyral Notebooks offer a space for teaching students to think-through digital text analysis and argue that the design of Spyral makes it rich in potential for the humanities researcher. We also highlight the collaborative possibilities of Spyral and show ways in which Spyral Notebooks have been used for collaborative digital humanities projects. Finally, we discuss future plans for Spyral.

## What are Spyral Notebooks?

Spyral ([voyant-tools.org/spyral](http://voyant-tools.org/spyral)) is a notebook development environment that is integrated into Voyant Tools ([voyant-tools.org](http://voyant-tools.org)). Notebook environments can be thought of as both extensions of traditional research notebooks and as novel tools that integrate documentation, active analysis and presentation of results. At their core, notebooks are made up of three types of blocks or cells that a user can add or delete in a sequence.

---

<sup>4</sup> See Welsh, Megan E. "Review of Voyant Tools." *Collaborative Librarianship* 6, no. 2 (2014): 96+.

1. There are **text cells** that can contain headings and other text elements found in word processors or browser editors (usually based in HTML) for typing unstructured text. Depending on the notebook environment, the text blocks can be simple or more sophisticated. Mathematica, another notebook environment, has a sophisticated text editing environment with integrated outlining and the ability to change stylesheets and generate slides from the text. Spyral Notebooks use HTML for text and offer an in-browser WYSIWYG HTML editor for the text blocks (see Figure 1).
2. There are **code cells** where the user inputs code, be it Python, the Wolfram language used in Mathematica, or JavaScript, which is used in Spyral. The code cells can be run in sequence or individually as you debug your code. Code cells can contain as much or as little code as the user desires (see Figure 1).
3. There are **output cells** which produce the output of the code you inputted in the associated code cell (see Figure 1). It is important to recognize that the output of the code is dependent on what you have instructed the computer to do; that is, it is not a printout of the code cell but the results of running your code. You thus have to instruct the computer to print out the desired results. In Figure 1, when you run the code cell Spyral outputs the traditional “Hello World.”

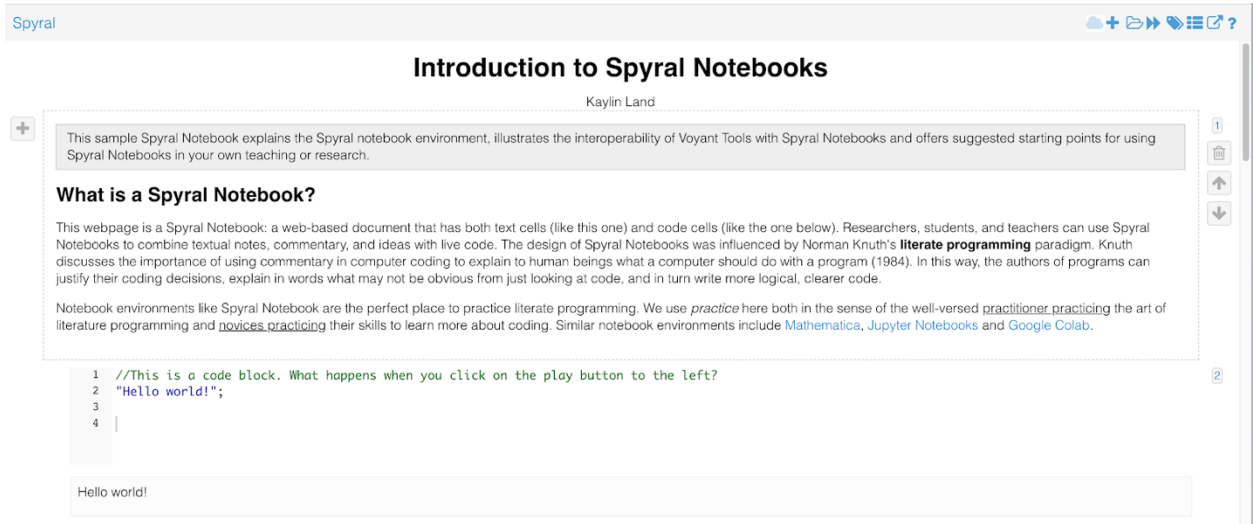


Figure 1 A sample Spyral Notebook showing a text block, code block and output block

Notebook environments are designed as “literate” programming environments. The term literate programming comes from Donald Knuth’s 1984 article of the same name. Knuth promoted a programming practice that encourages the human authors of code to notate code for other human readers to make clear what work the code is accomplishing. Knuth wrote: “Instead of imagining that our main task is to instruct a *computer* what to do, let us concentrate rather on explaining to *human beings* what we want a computer to do.”<sup>5</sup> Notebook environments encourage researchers to document their thinking in text blocks with accompanying code blocks that do the processing and analysis. In the case of Spyral, the text blocks also provide a rich opportunity for discussing the results of analysis and explaining what work code blocks are doing. For example, a code block that produces a visualization from Voyant Tools can be annotated with an explanation of that visualization and what decisions were made to generate such a visualization. This type of documentation is particularly beneficial in digital humanities

<sup>5</sup> Donald E. Knuth, “Literate Programming,” *The Computer Journal* 27, no. 2 (1984): 97-111.

projects where the textual accompaniments to code and visualizations can be just as important as the analysis itself.

Other notebook environments currently available include Wolfram Mathematica ([www.wolfram.com/mathematica](http://www.wolfram.com/mathematica)), Jupyter Notebooks ([jupyter.org](http://jupyter.org)), Observable ([observablehq.com](http://observablehq.com)), and Google Colab or Colaboratory ([colab.research.google.com](http://colab.research.google.com)). These notebook environments are designed primarily for data science, visualization and software engineering. Spyral is the first notebook environment developed specifically for digital text analysis and visualization.

Notebook environments can also be used effectively in teaching as tutorial text, example code and exercises can be woven together and run step by step as the student works through the notebook. One example of such tutorials is The Art of Literary Text Analysis, written in Jupyter Notebooks by Stéfan Sinclair and Geoffrey Rockwell.<sup>6</sup> This tutorial is currently being adapted for use in Spyral Notebooks as well. Another example of a tutorial notebook is William Turkel's textbook Digital Research Methods in Mathematica (2020).<sup>7</sup>

### Notebook Environment Design and Criticisms

Joel Grus of the Allen Institute for Artificial Intelligence recently (2018) gave an hour-long talk at JupyterCon entitled "I don't like notebooks."<sup>8</sup> In his talk, Grus outlines some common criticism of notebook environments. While notebook environments offer many benefits, they have been criticized both as tools for the experienced coder and as teaching tools. Grus argues that notebooks are difficult for beginning users, encourage bad habits and discourage

---

<sup>6</sup> See <https://github.com/sgsinclair/alta/blob/master/ipynb/ArtOfLiteraryTextAnalysis.ipynb>

<sup>7</sup> See <https://williamjturkel.net/digital-research-methods-with-mathematica/>

<sup>8</sup> Grus, Joel. "I hate notebooks." Filmed August 2018 at JupyterCon, New York, NY. Video, <https://www.youtube.com/watch?v=7jiPeIFXb6U>

good habits. He also points out that most notebooks do not have autocompletion or linting and that they are not necessarily effective for writing reproducible or sharable code. Another complaint Grus (and other researchers) has is that notebook tutorials are oftentimes not written to be effective teaching tools because they encourage clicking through code cells without necessarily understanding the content of those cells. Many of these complaints are addressed in academic papers about notebook environments. For example, Chattopadhyay et al. identify pain points in computational notebooks and come to some of the same conclusions as Grus.<sup>9</sup> Without defending notebook environments in general, this paper does engage some of the criticism of notebook environments as it relates to Spyral Notebooks.

Yihui Xie provides a useful distinction when considering the criticism of notebook environments in his blog post “The First Notebook War.”<sup>10</sup> Xie writes that notebook usage can be classified into two broad camps: notebooks used for software engineering and those used for data analysis. Xie argues that notebook environments are inherently better suited for data analysis than for software engineering. Furthermore, he views many of the criticisms of notebook environments as aimed at using notebooks for software engineering. When thinking about Spyral Notebooks, the primary audience for the notebooks is humanities researchers who will use the notebooks for textual analysis and will not be using notebook environments to write code that will eventually transition out of notebooks and into a product, as would a software engineer. In addition, many users of Spyral Notebooks will not consider themselves coders at all and thus Spyral as a notebook environment is not necessarily geared towards teaching general

---

<sup>9</sup> Souti Chattopadhyay, Ishita Prasad, Austin Z. Henley, Anita Sarma, and Titus Barik. “What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities.” In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (CHI '20). Association for Computing Machinery, New York, NY (2020), 1–12. DOI: <https://doi.org/10.1145/3313831.3376729>

<sup>10</sup> Yihui Xie, “The First Notebook War,” September 2019. <https://yihui.org/en/2018/09/notebook-war/>

coding skills but rather towards teaching textual analysis as performed with Voyant and through JavaScript in Spyral. Further, notebooks, as Knuth imagined, encourage the author to explain what they want to do and to document what worked and didn't. This places the emphasis on the thinking about texts done through the notebook not on the programming or opaque results. The intended audience is other humanists interested in following the research and analysis. The audience may not understand all the code, but the results are documented and can be replicated or adapted.

One of the biggest criticisms of notebook environments is that they are not intuitive for beginning users and that they require complicated set-up, particularly for those who are not already familiar with client-server models.<sup>11</sup> As Johnson recognizes, "When students download an instructor-prepared notebook, those not familiar with the client-server model...will initially be baffled by their inability to simply double-click on it to open it up."<sup>12</sup> This is one of Spyral's biggest advantages: no set-up is required to use Spyral Notebooks. The application is entirely web-based so the user need only open a web browser and enter the Spyral Notebook URL ([voyant-tools.org/spyral](http://voyant-tools.org/spyral)). Removing this barrier to entry means that users will not need to download and install anything to their personal computers and can begin to work in Spyral quickly. Remember that Spyral is an extension of Voyant.

Another issue Chattopadhyay et al. recognize with notebook environments is the difficulties around loading data into the notebook and the challenges of cleaning data.<sup>13</sup> Because Spyral Notebooks extend Voyant Tools, any textual corpus that has been uploaded to Voyant can easily be used in Spyral Notebooks using the `loadCorpus` JavaScript function (see Figure 2). The

---

<sup>11</sup> Chattopadhyay et al., "What's Wrong with Computational Notebooks?" 3.

<sup>12</sup> Johnson, "Benefits and Pitfalls of Jupyter Notebooks in the Classroom."

<sup>13</sup> Chattopadhyay et al., "What's Wrong with Computational Notebooks?" 3.

corpus object loaded into Spyral will have all the properties of a Voyant corpus, meaning it will already be tokenized.

If a user is already working in Voyant and sees some interesting results, there is an even easier way to load a corpus in Voyant, and that is to Export the tool and corpus to Spyral.

Choosing to Export “a new Spyral Notebook from this tool and data” creates a notebook with the code to load the corpus and output the tool with the current query.

Lack of linting and other support available in Integrated Development Environments (IDEs) is also cited as a limitation of some notebook environments. Spyral Notebooks do contain basic linting features as well as some basic error correction features (see Figure 3).

#### 1. Access a pre-existing Voyant corpus in Spyral

Use this code to take a pre-existing Voyant corpus and assign it a variable name in Spyral Notebooks. Here you call the loadCorpus method which signals to the computer to save your corpus as a special file type and assigns it to a variable name so you can access it again.

```
1 //Upload pre-existing Voyant corpus
2 loadCorpus('austen').assign('yourCorpus');
3 //replace 'austen' with your corpus ID and 'yourCorpus' with a name for your corpus
4
5 yourCorpus.text({limit: 100}); //prints first 1000 words of your corpus
```

```
LOVE AND FREINSHIP AND OTHER EARLY WORKS (Love And Friendshi --- LADY SUSAN by Jane Austen --- SENSE AND SENSIBILITY by Jane Austen --- PRIDE AND PREJUDICE By Jane Austen --- MANSFIELD PARK (1814) --- EMMA By Jane Austen --- NORTHANGER ABBEY by --- Persuasion by
```

Figure 2 A code cell showing how to load a Voyant corpus in Spyral



Figure 3 A code cell showing suggested inputs

## Sharing Spyral

Spyral Notebooks are designed to be cooperative and sharable. Spyral Notebooks can be easily shared through URLs. If a user wishes to make changes to a notebook, they can save the notebook under a new name, effectively forking the notebook. This makes Spyral Notebooks highly cooperative and eliminates many of the issues present in other notebook environments in



which users need to download dependencies and ensure that their personal machine environment matches the environment of the user who created the notebook. In simple terms, when you share a Spyral notebook, because the text corpus is saved on the Voyant Tools server, any user will be able to open your notebook and view your results without downloading any additional files.

Currently, a catalogue feature is in development for Spyral that will allow users to view notebooks they have created and find suggested notebooks, including tutorials (see Figure 4).

This will encourage researchers to collaborate as they will be able to view what other users have done in Spyral and even build upon pre-existing projects.<sup>14</sup>

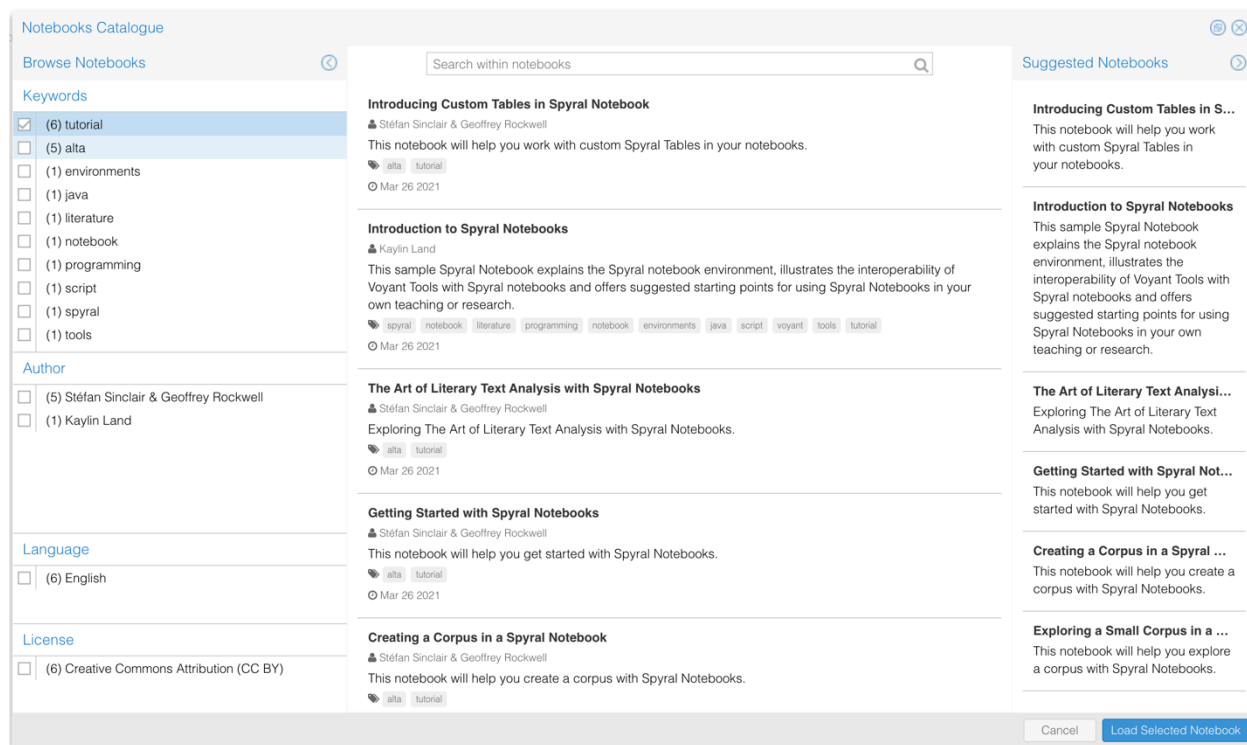


Figure 4 A screenshot of the Catalogue feature

In summary, Spyral Notebooks were designed for the humanities researcher with limited coding skills who wishes to experiment beyond the confines of Voyant. The notebooks provide a

---

<sup>14</sup> It should be noted that if a user wants to keep their notebook private they can install Voyant locally and then use Spyral locally too.

platform for collaborative projects and can be used both for teaching textual analysis and for saving, performing analysis, and presenting results.

### Digital Text Analysis in Spyral

Spyral Notebooks use JavaScript to extend the analytical suite available in Voyant Tools. The idea is to give digital humanists a growth path from what they can do in Voyant to their own analytics. To input text for analysis in Spyral, users can either reference an existing Voyant corpus or create a new corpus from strings, URLs, spreadsheets, or by uploading text files directly. Spyral Notebooks offer far more customizability for corpora than the upload button in Voyant Tools and also enable the user to edit their corpus after it has been uploaded (for example, to add more texts to a corpus). Once a corpus has been created, the user can call up any of the tools available in Voyant (see Figure 5). The user can then customize which features they want the tool to display by using different configuration options. For example, the user can call up the Cirrus word cloud visualization and change the background colour, font, limit the number of terms to load, edit the stop list and other features. These customization features are available in Voyant, but by accessing the visualization through Spyral the user can save the customization options and repeat them across different tools or corpora.



example, you can create a positive and negative category and perform sentiment analysis on your text.



Figure 6 A chart made from a customized table

## Spyral as a Pedagogical Tool

Spyral Notebooks were also designed with learning and teaching in mind. As many users of Voyant Tools come to the platform with limited coding skills, Spyral can be seen as a next step for learning digital text analysis that focuses on the thinking-through that goes into analysis. The platform is designed to be accessible for those users who are already familiar with JavaScript. They will be able to quickly harness the power of Spyral by reading the documentation (available at <https://voyant-tools.org/docs/#!/api>).

More importantly, Spyral is designed to provide a path from using a tool like Voyant to documenting text analysis to programming text analysis for those who have no experience with data analysis and coding languages. As mentioned above one can generate the code for particular configurations of Voyant tools and texts by Exporting to Spyral. Then one can add text cells

explaining and reflecting on the results generated. Thus one can document a project without learning more than how Export code. This can be shared with others for comment. It isn't a big step then to editing the code generated, perhaps to change the query or some of the parameters. One might then look at other people's notebooks and copy useful code. One can thus learn by remixing.

Currently, the first chapters of *The Art of Literary Text Analysis* are available for Spyral Notebooks and they walk users through the basics of creating a corpus in Spyral, exploring a small corpus and creating tables in Spyral (<https://voyant-tools.org/spyral/homeALTA>). This tutorial is designed to introduce users not only to working in Spyral but also to the basic tenets of literary text analysis. This makes it distinctive from other notebook tutorials that aim to teach coding basics without showing how those tools can be utilized for practical applications. Furthermore, the tutorial is focused primarily on helping students get started right away and teaches only the elements of JavaScript needed to perform textual analysis.

We are currently in the process of developing a collection of JavaScript recipes for use in Spyral Notebooks that can serve as an intermediary between a fully-fledged tutorial and the existing documentation for Spyral (<https://voyant-tools.org/spyral/JSrecipes#1>). The existing documentation is based on JSDoc (<https://jsdoc.app/>) and, while thoroughly documented, presents a high barrier to entry for the novice user. The JS Recipes for Spyral explain in prose how the documentation can be used and show users where to edit the code with their own variables. In this way, it will enable those with a clear end goal (for example, to upload a corpus using URLs) to quickly identify, copy, and import existing code into a Spyral notebook. With Spyral, it is simple to export single code blocks from one notebook to another, meaning that users will not be reliant on copy and paste to share code from notebook to notebook.

## Spyral as a Collaborative Tool

Spyral Notebooks also offer collaborative capabilities for humanities researchers to work together. As discussed above, Spyral Notebooks are easily sharable via URLs and can be edited if users share the editing passcode required to save the notebook under the same URL. Because Spyral Notebooks do not require any downloading, researchers do not need to worry about ensuring that their computers have the same dependencies when loading the notebooks. This means that any researcher will be able to view the notebook the same way as the notebook's creator. In addition, researchers who are more knowledgeable about coding will be able to add or edit existing code blocks and researchers who may be less familiar with code can view the outputs without needing to understand exactly how the code is functioning.

Currently we are in the process of developing further collaborative tools for Spyral, including the Catalogue view that will allow researchers to easily find and utilize existing notebooks in their own projects. We are also developing new features with support from the LINCS project ([lincsproject.ca/](http://lincsproject.ca/)) led by Susan Brown and supported by CFI. Spyral is being enhanced by researchers who want to use linked open data to enhance their analysis of texts.

## Further Directions

Spyral is still being developed. Current limitations of Spyral include lack of security features beyond password protecting notebooks as well as challenges scaling notebooks to include large amounts of data. We envision for Spyral Notebooks a comprehensive environment that can be used equally for introducing programming, teaching digital text analysis and for performing data analysis. We invite you to try out Spyral Notebooks and contact us with any questions or comments you have about the tool.

## Works Cited

Chattopadhyay, Souti, Ishita Prasad, Austin Z. Henley, Anita Sarma, and Titus Barik. “What’s

Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities.”

*Conference on Human Factors in Computing Systems - Proceedings*, 2020, 1–12.

<https://doi.org/10.1145/3313831.3376729>.

Grus, Joel. “I hate notebooks.” Filmed August 2018 at JupyterCon, New York, NY. Video,

<https://www.youtube.com/watch?v=7jiPeIFXb6U>

Johnson, Jeremiah W. “Benefits and Pitfalls of Jupyter Notebooks in the Classroom.” *SIGITE*

*2020 - Proceedings of the 21st Annual Conference on Information Technology Education*,

2020, 32–37. <https://doi.org/10.1145/3368308.3415397>.

Knuth, Donald E. “Literate Programming.” *The Computer Journal* 27, no. 2 (1984): 97-111.

Xie, Yihui. “The First Notebook War,” September 2019. <https://yihui.org/en/2018/09/notebook-war/>